Working Draft American National Standard

Project T10/1528-D

Revision 00
21 September 2004

Information technology - SCSI Bridge Controller Commands (BCC)

T10 Technical Editor: Robert C Elliott
Hewlett-Packard Corporation
MC 140801
PO Box 692000
Houston, TX 77269-2000
USA

Telephone: 281-518-5037
Email: elliott@hp.com

Reference number
ISO/IEC 14776-511:200x
ANSI INCITS.*:200x**

Printed 5:10 PM Tuesday 21 September 2004

Points of contact

<u>International Committee for Information Technology Standards (INCITS) T10 Technical Committee</u>

<u>T10 Chair</u>
John B. Lohmeyer
LSI Logic
4420 Arrows West Drive
Colorado Springs, CO 80907-3444
USA

Telephone: (719) 533-7560
Email: lohmeyer@t10.org

<u>T10 Vice-Chair</u>
George O. Penokie
IBM Corporation
MS: 2C6
3605 Highway 52 N
Rochester, MN 55901
USA

Telephone: (507) 253-5208
Email: gop@us.ibm.com

T10 Web Site: http://www.t10.org

<u>T10 E-mail reflector:</u>
Server: majordomo@t10.org
To subscribe send e-mail with ‘subscribe’ in message body
To unsubscribe send e-mail with ‘unsubscribe’ in message body

<u>INCITS Secretariat</u>
Suite 200
1250 Eye Street, NW
Washington, DC 20005
USA

Telephone: 202-737-8888
Web site: http://www.incits.org
Email: incits@itic.org

<u>Information Technology Industry Council</u>
Web site: http://www.itic.org

<u>Document Distribution</u>
INCITS Online Store
managed by Techstreet
1327 Jones Drive
Ann Arbor, MI 48105
USA

Web site: http://www.techstreet.com/incits.html
Telephone: (734) 302-7801 or (800) 699-9277

Global Engineering Documents, an IHS Company
15 Inverness Way East
Englewood, CO 80112-5704
USA

Web site: http://global.ihs.com
Telephone: (303) 397-7956 or (303) 792-2181 or (800) 854-7179

American National Standard
for Information Technology

SCSI Bridge Controller Commands (BCC)

Secretariat
Information Technology Industry Council

Approved mm.dd.yy

American National Standards Institute, Inc.

ABSTRACT

This standard specifies the functional requirements for the SCSI Block Commands - 2 (SBC-2) command set. SBC-2 permits SCSI block logical units such as rigid disks to attach to computers and provides the definition for their use.

This standard maintains a high degree of compatibility with the SCSI Block Commands (SBC) command set, INCITS 306-1998, and while providing additional functions, is not intended to require changes to presently installed devices or existing software.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by
American National Standards Institute
11 W. 42nd Street, New York, New York 10036

Printed in the United States of America

Revision Information

R.1 Revision 0 (21 September 2004)

Incorporated the following proposal per the May 2004 CAP WG (04-134r0) and T10 plenary (04-135r1).

a) 03-364r1 MSC Report Bridge Mapping command (Rob Elliott, HP)

Renamed the standard from Management Server Commands (MSC) to Bridge Controller Commands (BCC) per the September 2004 T10 plenary (04-289r0).

Contents

Page

Tables

Figures

Foreword (This foreword is not part of this standard)

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, International Committee for Information Technology Standards, Information Technology Institute, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by the International Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, INCITS had the following members:

Karen Higginbottom, Chair

David Michael, Vice-Chair

INCITS Technical Committee T10 on Lower Level Interfaces, which developed and reviewed this standard, had the following members:

Introduction

The standard is organized as follows:

Clause 1 (Scope) describes the relationship of this standard to the SCSI family of standards.
Clause 2 (Normative References) provides references to other standards and documents.
Clause 3 (Definitions, symbols, abbreviations, keywords, and conventions) defines terms and conventions used throughout this standard.
Clause 4 (Bridge controller device type model) provides an overview of the bridge controller device class and the command set.
Clause 5 (Commands for bridge controller devices) defines commands specific to bridge controller devices.
Clause 6 (Parameters for bridge controller devices) defines diagnostic pages, mode parameters and pages, log pages, and VPD pages specific to bridge controller devices.

American National Standard
for Information Technology -

SCSI Bridge Controller Commands (BCC)

1 Scope

This standard defines the command set extensions to facilitate operation of SCSI bridge controller devices. The clauses of this standard, implemented in conjunction with the applicable clauses of SPC-3, fully specify the standard command set for SCSI bridge controller devices.

The objective of this standard is to:

a) permit an application client to communicate over a SCSI service delivery subsystem with a logical unit that declares itself to be a bridge controller device in the PERIPHERAL DEVICE TYPE field of the standard INQUIRY data (see SPC-3);
b) define the addressing model to address bridge controller device logical units in the path between an initiator port and target port; and
c) define commands unique to the bridge controller device type.

Figure 1 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards.

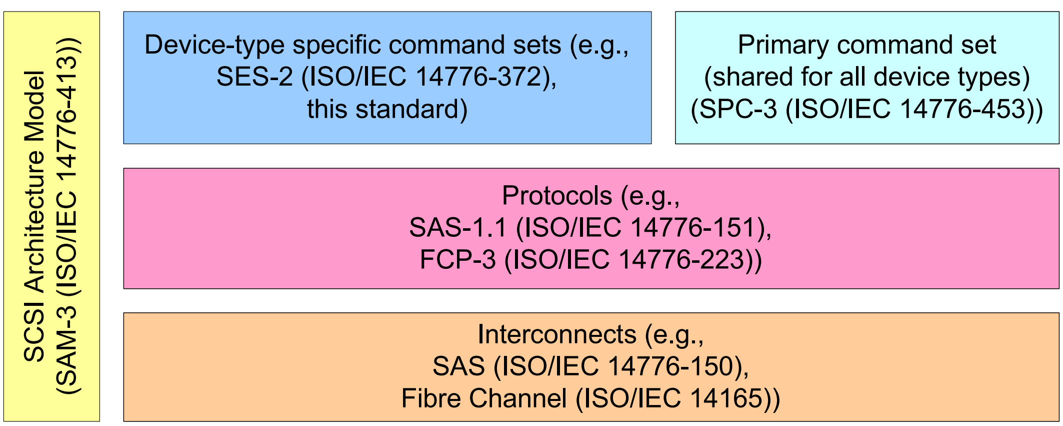


Figure 1 — SCSI document relationships

Figure 1 is intended to show the general relationship of the documents to one another, and is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture.

The set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

2 Normative References

2.1 Normative references overview

The following standards contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI:

a) approved ANSI standards;
b) approved and draft international and regional standards (e.g., ISO, IEC, CEN/CENELEC, ITU-T); and
c) approved and draft foreign standards (e.g., BSI, JIS, and DIN).

For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

Additional availability contact information is provided below as needed.

Table 1 lists standards bodies and their web sites.

Table 1 — Standards bodies

Abbreviation	Standards body	Web site
ANSI	American National Standards Institute	http://www.ansi.org
BSI	British Standards Institution	http://www.bsi-global.com
CEN	European Committee for Standardization	http://www.cenorm.be
CENELEC	European Committee for Electrotechnical Standardization	http://www.cenelec.org
DIN	German Institute for Standardization	http://www.din.de
IEC	International Engineering Consortium	http://www.iec.ch
IEEE	Institute of Electrical and Electronics Engineers	http://www.ieee.org
IETF	Internet Engineering Task Force	http://www.ietf.org
INCITS	International Committee for Information Technology Standards	http://www.incits.org
ISO	International Standards Organization	http://www.iso.ch
ITI	Information Technology Industry Council	http://www.itic.org
ITU-T	International Telecommunications Union Telecommunications Standardization Sector	http://www.itu.int
JIS	Japanese Industrial Standards Committee	http://www.jisc.org
T10	INCITS T10 Committee - SCSI storage interfaces	http://www.t10.org
T11	INCITS T11 Committee - Fibre Channel interfaces	http://www.t11.org
T13	INCITS T13 Committee - ATA storage interface	http://www.t13.org

2.2 Approved references

At the time of publication, the following referenced standards were approved.

ISO/IEC 14776-342, *SCSI-3 Controller Commands - 2 (SCC-2)*(ANSI INCITS 318-1998)
RFC 3720, *Internet Small Computer Systems Interface (iSCSI)*

2.3 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the documents, or regarding availability, contact the relevant standards body as indicated.

ISO/IEC 14776-413, *SCSI Architecture Model - 3 (SAM-3) standard* (T10/1561-D)
ISO/IEC 14776-453, *SCSI Primary Commands - 3 (SPC-3) standard* (T10/1416-D)
ISO/IEC 14776-372, *SCSI Enclosure Services - 2 (SES-2) standard* (T10/1559-D)

NOTE 1 - For more information on the current status of the document, contact the INCITS Secretariat at 202-737-8888 (telephone), 202-638-4922 (fax) or via Email at incits@itic.org. To obtain copies of this document, contact Global Engineering at 15 Inverness Way East Englewood, CO 80112-5704 at 800-854-7179 (telephone), 303-792-2181 (telephone), or 303-792-2192 (fax).

3 Definitions, symbols, abbreviations, keywords, and conventions

3.1 Definitions

3.1.1 additional sense code: A combination of the ADDITIONAL SENSE CODE and ADDITIONAL SENSE CODE QUALIFIER fields in the sense data. See SPC-3.

3.1.2 application client: An object that is the source of SCSI commands. See SAM-3.

3.1.3 byte: A sequence of eight contiguous bits considered as a unit.

3.1.4 command: A request describing a unit of work to be performed by a device server. See SAM-3.

3.1.5 command descriptor block (CDB): The structure used to communicate commands from an application client to a device server. See SPC-3.

3.1.6 data-in buffer: The buffer identified by the application client to receive data from the device server during the processing of a command. See SAM-3.

3.1.7 data-out buffer: The buffer identified by the application client to supply data that is sent from the application client to the device server during the processing of a command. See SAM-3.

3.1.8 device server: An object within a logical unit that processes SCSI tasks according to the rules of task management. See SAM-3.

3.1.9 device type: The type of device (or device model) implemented by the device server as indicated by the PERIPHERAL DEVICE TYPE field of the standard INQUIRY data. See SPC-3.

3.1.10 direct-access block device: A device that is capable of containing data stored in blocks that each have a unique logical block address.

3.1.11 domain: An I/O system consisting of a set of SCSI devices that interact with one another by means of a service delivery subsystem. See SAM-3.

3.1.12 field: A group of one or more contiguous bits, a part of a larger structure such as a CDB (see 3.1.5) or sense data (see SPC-3).

3.1.13 hard reset: A condition resulting from the events defined by SAM-3 in which the SCSI device performs the hard reset operations described in SAM-3, SPC-3, SES-2 (if applicable), and this standard.

3.1.14 I_T nexus loss: A condition resulting from the events defined by SAM-3 in which the SCSI device performs the I_T nexus loss operations described in SAM-3, SPC-3, SES-2 (if applicable), and this standard.

3.1.15 logical unit (LU): An externally addressable entity within a target that implements a SCSI device model and contains a device server. A detailed definition of a logical unit may be found in SAM-3.

3.1.16 logical unit number (LUN): An encoded 64-bit identifier for a logical unit. A detailed definition of a logical unit number may be found in SAM-3.

3.1.17 logical unit reset: A condition resulting from the events defined by SAM-3 in which the logical unit performs the logical unit reset operations described in SAM-3, SPC-3, SES-2 (if applicable), and this standard.

3.1.18 power cycle: Power being removed followed by power being applied to a SCSI device.

3.1.19 power on: A condition resulting from the events defined by SAM-3 in which the SCSI device performs the power on operations described in SAM-3, SPC-3, SES-2 (if applicable), and this standard.

3.1.20 sense data: Data describing an error or exceptional condition that a device server delivers to an application client in association with CHECK CONDITION status. See SPC-3.

3.1.21 sense key: The contents of the SENSE KEY field in the sense data. See SPC-3.

3.1.22 status: One byte of response information sent from a device server to an application client upon completion of each command. See SAM-3.

3.1.23 well-known logical unit (W-LU): A logical unit that only does specific functions (see clause 8). Well known logical units allow an application client to issue requests to receive and manage specific information usually relating to a SCSI target.

3.1.24 well-known logical unit number (W-LUN): The logical unit number that identifies a well known logical unit.

3.2 Symbols and abbreviations

See table 1 for abbreviations of standards bodies (e.g., ISO). Additional symbols and abbreviations used in this standard include:

Abbreviation	**Meaning**
CDB	command descriptor block (see 3.1.5)
FCP	Fibre Channel Protocol (revision not relevant)
FCP-3	Fibre Channel Protocol - 3 standard
I/O	input/output
iSCSI	Internet SCSI standard
LSB	least significant bit
LU	logical unit (see 3.1.15)
LUN	logical unit number (see 3.1.16)
MSB	most significant bit
SAM-3	SCSI Architecture Model - 3 standard
SAS	Serial Attached SCSI (revision not relevant)
SAS-1.1	Serial Attached SCSI - 1.1 standard
SCSI	Small Computer System Interface family of standards
SCC-2	SCSI-3 Controller Commands - 2 standard
SES-2	SCSI Enclosure Services - 2 standard
SPC-3	SCSI Primary Commands - 3 standard
W-LU	well-known logical unit (see 3.1.23)
W-LUN	well-known logical unit number (see 3.1.24)

3.3 Keywords

3.3.1 can: A keyword used for statements of possibility and capability indicating a condition that is required to be handled (equivalent “it is possible to”).

3.3.2 cannot: A keyword used for statements of possibility and capability indicating a condition that is not required to be handled (equivalent “it is not possible to”).

NOTE 2 - “May” signifies permission expressed by this standard, whereas “can” refers the ability of a device compliant with this standard to handle events outside of control of this standard.

3.3.3 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.3.4 ignored: A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value shall not be examined by the receiving SCSI device and may be set to any value by the transmitting SCSI device.

3.3.5 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as an error.

3.3.6 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.3.7 may: A keyword that indicates flexibility of choice with no implied preference; equivalent to “may or may not” and equivalent to the phrase “it is permitted.”

3.3.8 may not: Keywords that indicate flexibility of choice with no implied preference; equivalent to “may or may not” and equivalent to the phrase “it is permitted.”

3.3.9 need not: Keywords indicating a feature that is not required to be implemented; equivalent to “is not required that.”

3.3.10 obsolete: A keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard.

3.3.11 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

3.3.12 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

3.3.13 restricted: A keyword referring to bits, bytes, words, and fields that are set aside for use in other SCSI standards. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this standard.

3.3.14 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.15 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase “it is strongly recommended.”

3.3.16 vendor-specific: Something (e.g., a bit, field, or code value) that is not defined by this standard and may be used differently in various implementations.

3.4 Conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in this clause or in the text where they first appear.

Names of commands, status codes, sense keys, and additional sense codes are in all uppercase (e.g., REQUEST SENSE).

Names of fields and state variables are in small uppercase (e.g. NAME). When a field or state variable name contains acronyms, uppercase letters may be used for readability. Normal case is used when the contents of a field or state variable are being discussed. Fields or state variables containing only one bit are usually referred to as the NAME bit instead of the NAME field.

Normal case is used for words having the normal English meaning.

The ISO convention of numbering is used (i.e., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). Table 2 shows a comparison of the ISO and American numbering conventions.

Table 2 — ISO and American numbering conventions

ISO	American
0,6	0.6
3,141 592 65	3.14159265
1 000	1,000
1 323 462,95	1,323,462.95

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (e.g., 0101b) are binary values. Underscores may be included in binary values to increase readability or delineate field boundaries (e.g., 0101_1010b).

A sequence of numbers or upper case letters ‘A’ through ‘F’ immediately followed by lower-case h (e.g., FA23h) are hexadecimal values. Underscores may be included in hexadecimal values to increase readability or delineate field boundaries (e.g., FD8C_FA23h).

Lists sequenced by letters (e.g., a) red, b) blue, c) green) show no ordering relationship between the listed items. Numbered lists (e.g., 1) red, 2) blue, 3) green) show an ordering between the listed items.

If a conflict arises between text, tables or figures, the order of precedence to resolve the conflicts is text, then tables, and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values.

Notes do not constitute any requirements for implementers.

4 Bridge controller device type model

4.1 Bridge controller device type model overview

Bridge controller devices (e.g., gateways, routers) enable communication between SCSI devices in two or more SCSI domains, providing initiator ports in one domain access to target ports in the other domain(s). The SCSI domains usually support different SCSI transport protocols (e.g., Fibre Channel (see FCP-3), Serial Attached SCSI (see SAS-1.1), TCP/IP (see iSCSI), or InfiniBand (see SRP-2)).

Bridges may be designed to be invisible, so that application clients using initiator ports in one domain see the target ports as native target ports in that domain and do not know that a bridge is present. This approach generally works for simple read and write commands, but often fails to support more complicated SCSI commands like the EXTENDED COPY command (see SPC-3).

If the bridge is paired with a SCSI target device with a known command set (e.g., the bridge fronts a SAS tape drive, providing it with an FCP interface), the manufacturer is able to ensure that the bridge supports all the commands that the SCSI target device supports, making this limitation irrelevant. However, general-purpose bridges (e.g., TCP/IP to Fibre Channel gateways or routers) that support arbitrary SCSI devices can encounter new device types or new commands for which they were not designed.

This standard allows bridges to be visible to application clients, exposing the transport protocol translations that are occuring and reducing or eliminating the need for bridges to intercept and change data (e.g., the application client is capable of understanding why the protocol identifier returned in an INQUIRY command VPD page doesn’t match the transport protocol its initiator port uses, and the bridge is not required to intercept and change that VPD page parameter data to translate the protocol identifier).

NOTE 3 - Although this information may be available via out-of-band management interfaces such as the Storage Networking Industry Association (SNIA)’s Storage Management Interface Specification (SMI-S), the SCSI application client may not have access to that information. However, the application client should be able to generate SCSI commands to the bridge if it is able to generate SCSI commands that are mapped through a bridge.

This standard is intended to be used in conjunction with SAM-3, SPC-3, and SES-2.

Figure 2 shows an example TCP/IP (i.e., iSCSI) to Fibre Channel (i.e., FCP) bridge, showing initiator port, target port, and logical unit objects. The initiator ports all use iSCSI and the target ports all use FCP in this example.

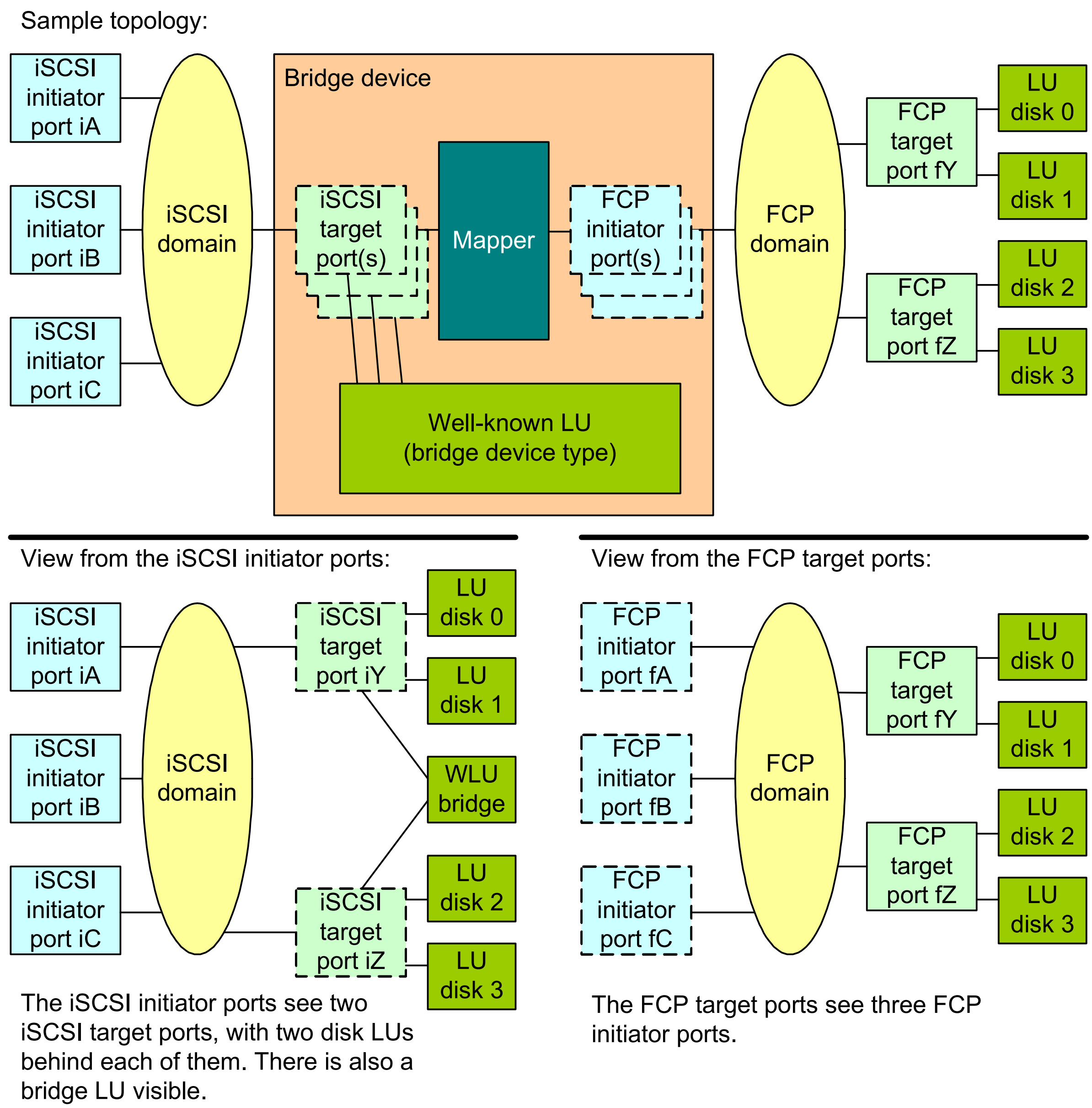


Figure 2 — iSCSI to FCP bridge example

To enable communication between the iSCSI initiator ports and the FCP target ports, the bridge maps:

a) the FCP target ports into iSCSI target ports in the iSCSI domain (target mapping); and
b) the iSCSI initiator ports into FCP initiator ports in the FCP domain (initiator mapping).

Each FCP target port receives requests and delivers responses to what it sees as an FCP initiator port but is really the bridge acting on behalf of an iSCSI initiator port. Similarly, each iSCSI initiator port sends requests and receives responses from what it sees as an iSCSI target port but is really the bridge acting on behalf of an FCP target port.

The bridge interprets the requests and responses (e.g., the frames transmitting commands, task management functions, data, and status) received from one domain and recreates them on the other domain with modifications based on the mapper (e.g., changing the initiator port identifier, target port identifier, logical unit number, and task tag).

Initiator ports and target ports may be on located on either side of the bridge. Figure 3 shows an example with multiple iSCSI initiator ports, one iSCSI target port, one FCP initiator port, and multiple FCP target ports.

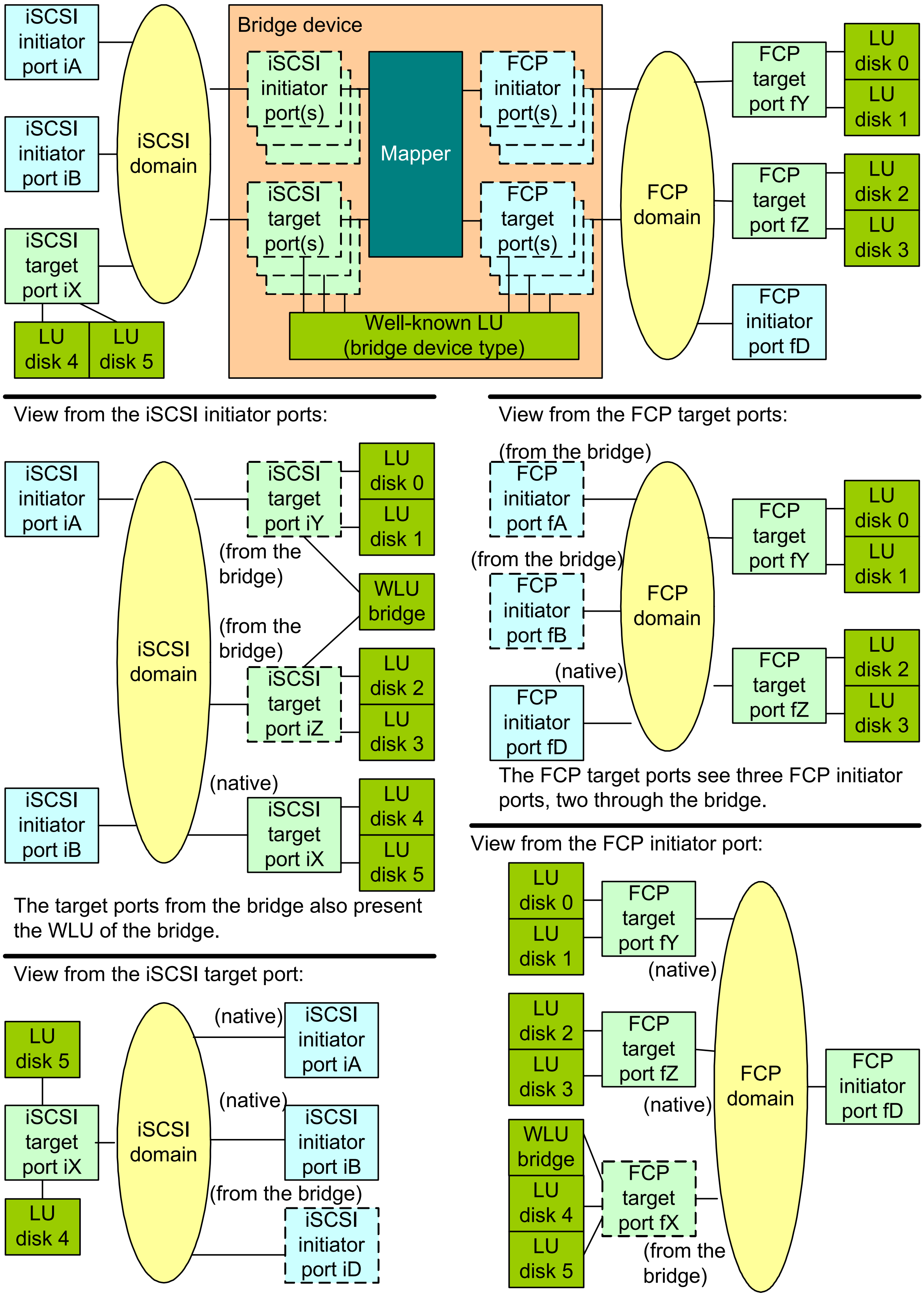


Figure 3 — iSCSI to FCP bridge example with initiator ports and target ports on both sides

In most protocols, the bridge need not employ lots of physical ports to serve these distinct roles; it shares a small number of physical ports whose number depends on performance and reliability requirements. The techniques available to the bridge to share physical ports differs for each protocol:

a) In Fibre Channel (see FCP-3), the bridge may:
   A) use virtual N_Ports to let the bridge appear to be multiple SCSI ports at once;
   B) use a virtual loop to let the bridge appear as multiple SCSI ports; or
   C) emulate a switch with multiple SCSI ports behind it;
b) In TCP/IP (see iSCSI), the bridge may establish many SCSI ports at an IP address;
c) In InfiniBand (see SRP-2), the bridge may establish many SCSI ports behind a channel adapter; and
d) In Serial Attached SCSI (see SAS-1.1), the bridge may appear as an expander with multiple SCSI ports attached to it.

4.2 Mapping

4.2.1 Mapping logical units

Bridges vary in complexity and capability. For mapping of logical units behind target ports on the far side (i.e., in a different domain than the initiator port) of a bridge to the near side (i.e., to the same domain as the initiator port), two approaches are:

a) **Target port mapping**. Each target port on the far side of the bridge is mapped to a unique target port on the near side of the bridge. Logical unit numbers accessible via each target port are unchanged, even though the protocol of the target port may change. This may be difficult to implement on some protocols because the bridge has to pretend to have virtual target ports even though it might only have one physical port.
b) **Logical unit mapping**. The bridge presents one or more target ports on the near side of the bridge, but there is not a one-to-one correspondence with the target ports on the far side of the bridge. Select logical units from select target ports on the far side of the bridge are mapped to LUNs behind each near side target port. As far as the initiator port is concerned, the bridge serves as the SCSI target device containing those logical units; the fact that they are really in separate SCSI target devices on the far side is hidden.

   Due to the likely LUN collisions (many of the back-side LUNs are probably LUN 0), the bridge must remap all the LUNs. Since combining them into one target device and renumbering them changes the target port/logical unit relationship, it introduces several problems including:

   A) REPORT LUNS data from any logical unit is wrong. A logical unit only knows about logical units in the same physical SCSI target device. It doesn’t know about other logical units that the bridge has mapped as its peers. The logical unit doesn’t even know its own LUN as seen through the bridge. The bridge must intercept REPORT LUNS data and change it to reflect the target device LUN inventory that the bridge has created;
   B) Commands using relative target port identifiers reference the wrong values. The far side target device’s relative target port identifiers are different than the near side target device’s relative target port identifiers. In particular, the target port group access commands (i.e., the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS command) do not work correctly. It may be difficult for a bridge to intercept these commands and modify them to behave correctly; and
   C) Well-known logical unit numbers conflict with each other. If the far side target device has any well-known logical units, it is impossible for the bridge to map them and preserve their well-known LUNs. They can overlap with each other and with the bridge’s own well-known logical units (e.g., nested bridges well-known logical units conflict).

4.2.2 Mapping initiator ports

The bridge should map each initiator port to a unique far side initiator port. For many SCSI commands, the target port needs to realize there are separate initiator ports communicating with it (e.g., this is critical for access controls and persistent reservations)

Some bridges are only capable of presenting one initiator port on the far side. Commands cognizant of initiator port identity like access controls and persistent reservations do not operate correctly unless the bridge intercepts them and implements them itself (i.e., at least for those commands, the bridge serves as a full-fledged SCSI target port and device server and performs the commands itself without forwarding them to the far side target ports). This standard allows these bridges to identify themselves and identify which commands they support so application clients may determine when not to send certain commands through them.

4.3 Addressing

4.3.1 Bridge controller well-known LUN

Any logical unit in a bridge controller device may report a peripheral device type of <TBD; probably 10h which is currently reserved for Bridging Expanders> in its standard INQUIRY data indicating that it is a bridge controller.

Editor's Note 1: Assignment of the peripheral device type for bridge controller devices must be made in SPC-4.

A bridge controller shall provide access to a bridge controller logical unit via the well-known logical unit number (W-LUN) of <TBD; maybe 04h>.

Editor's Note 2: Assignment of the W-LUN must be coordinated with SPC-4, the only other standard that has assigned W-LUNs so far.

4.3.2 Bridge addressing method

SAM-3 defines an 8-byte hierarchical LUN format composed of four 2-byte addressing fields, with the format of each 2-byte addressing field specified by the ADDRESS METHOD field in the first byte. The LUN is used to identify the nexus used to route requests and responses for commands and task management functions. How it is used is transport protocol specific (e.g., some protocols include a LUN field in frames delivering command frames and task management frames, but do not include it in frames transferring data or status).

Table 3 shows the format with the ADDRESS METHOD field set to 10b (i.e., the logical unit addressing format) defined in SAM-3.

Table 3 — Logical unit addressing (informative)

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (10b)		TARGET					
1	BUS NUMBER			LUN				

Bridge controllers interpret addressing fields with the ADDRESS METHOD field set to 10b differently than other device types. Table 4 defines the bridge addressing format.

Table 4 — Bridge addressing

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (10b)		(MSB)					
1	RELATIVE INITIATOR PORT							(LSB)

The RELATIVE INITIATOR PORT field specifies the relative port identifier of the initiator port that the bridge shall use to forward the request or response.

This does not conflict with other device type's uses of the logical unit addressing format. If a bridge receives a LUN containing an ADDRESS METHOD field set to 10b, it interprets it per table 4. If logical unit with another device type (e.g., SCC-2) receives a LUN containing an ADDRESS METHOD field set to 10b, it interprets it per table 3.

Each bridge interprets the FIRST LEVEL ADDRESSING field (i.e., the first 2 bytes) of the LUN to decide which of its initiator ports to use. When it forwards the incoming request or response, it shall create a new LUN derived from the incoming LUN by setting:

a) the FIRST LEVEL ADDRESSING field to the contents of the incoming SECOND LEVEL ADDRESSING field;
a) the SECOND LEVEL ADDRESSING field to the contents of the incoming THIRD LEVEL ADDRESSING field;
a) the THIRD LEVEL ADDRESSING field to the contents of the incoming FOURTH LEVEL ADDRESSING field; and
b) the FOURTH LEVEL ADDRESSING field to 0000h.

4.3.3 Addressing examples

To access W-LU Z in figure 3 (see 4.1), iSCSI initiator port iA, iB, or iC sends a command to target port identifier iF, iG, iH, iI, or iJ with the LUN shown in table 5. It does not use iD or iE to access W-LU Z, since those target ports are not behind the first bridge.

Table 5 — Accessing W-LU Z from iA, iB, or iC - initial (and only) LUN

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (11b)		LENGTH		EXTENDED ADDRESS METHOD (1h)			
1	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Z)							(LSB)
2	Unused (000000h)							
3								
4								
5								
6								
7								

To access W-LU Y in the second bridge, iSCSI initiator port iA, iB, or iC sends a command to target port identifier iH, iI, or iJ with the LUN shown in table 6. It does not use target ports iD, iE, iF, or iG, since they are not behind the second bridge. This is a two-level LUN. It instructs the first bridge which output port to use to send the rest of the LUN (shifted into a first-level LUN).

Table 6 — Accessing W-LU Y from iA, iB, or iC - initial LUN

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (10b)		(MSB)					
1	RELATIVE INITIATOR PORT (in first bridge)							(LSB)
2	ADDRESS METHOD (11b)		LENGTH		EXTENDED ADDRESS METHOD (1h)			
3	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Y)							(LSB)
4	Unused (0000h)							
5								
6								
7								

As the first bridge forwards this LUN, it shifts off the first two bytes and sets the last two bytes to zero, sending the command to the target port identifier fH, fI, or fJ based on the first bridge's mapper with the LUN changed as shown in table 7.

Table 7 — Accessing W-LU Y from iA, iB, or iC- LUN as output by first bridge

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (11b)		LENGTH		EXTENDED ADDRESS METHOD (1h)			
1	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN Y)							(LSB)
2	Unused (000000h)							
3								
4								
5								
6								
7								

If the first bridge does not have a map yet for target ports fH, fI, or fJ, W-LU Y is inaccessible. Bridges should include a target port of their own available for management; if that target port is mapped, communication may occur to the W-LU.

Figure 4 shows the target port identifiers and LUN values described in table 5, table 6, and table 7 as they are used to access the bridges also shown in figure 4.

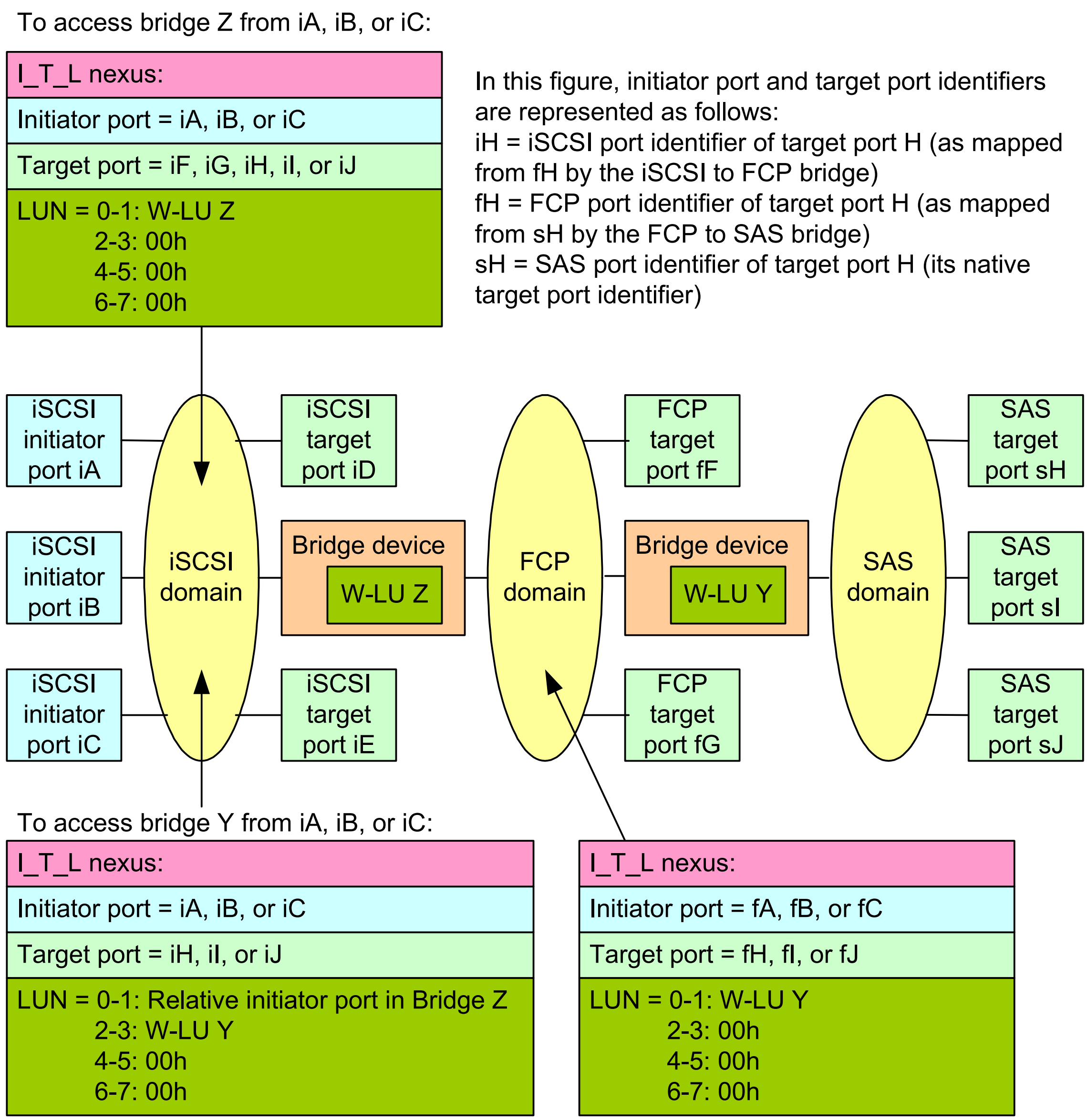


Figure 4 — Accessing W-LU Z and W-LU Y from iA, iB, and iC

4.3.4 Maximum bridge addressing LUN example

The 8-byte hierarchical LUN format supports a maximum of four levels of bridges as shown in table 8. Larger topologies are not accessible through the 8-byte LUN field and are beyond the scope of this standard.

Table 8 — Maximum bridge addressing LUN

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (10b)		(MSB)					
1	RELATIVE INITIATOR PORT (in first bridge)							(LSB)
2	ADDRESS METHOD (10b)		(MSB)					
3	RELATIVE INITIATOR PORT (in second bridge)							(LSB)
4	ADDRESS METHOD (10b)		(MSB)					
5	RELATIVE INITIATOR PORT (in third bridge)							(LSB)
6	ADDRESS METHOD (11b)		LENGTH		EXTENDED ADDRESS METHOD (1h)			
7	EXTENDED ADDRESS METHOD SPECIFIC (W-LUN of fourth level bridge)							(LSB)

4.3.5 Example of peer bridges

Figure 5 shows an example with both nesting and peer bridges.

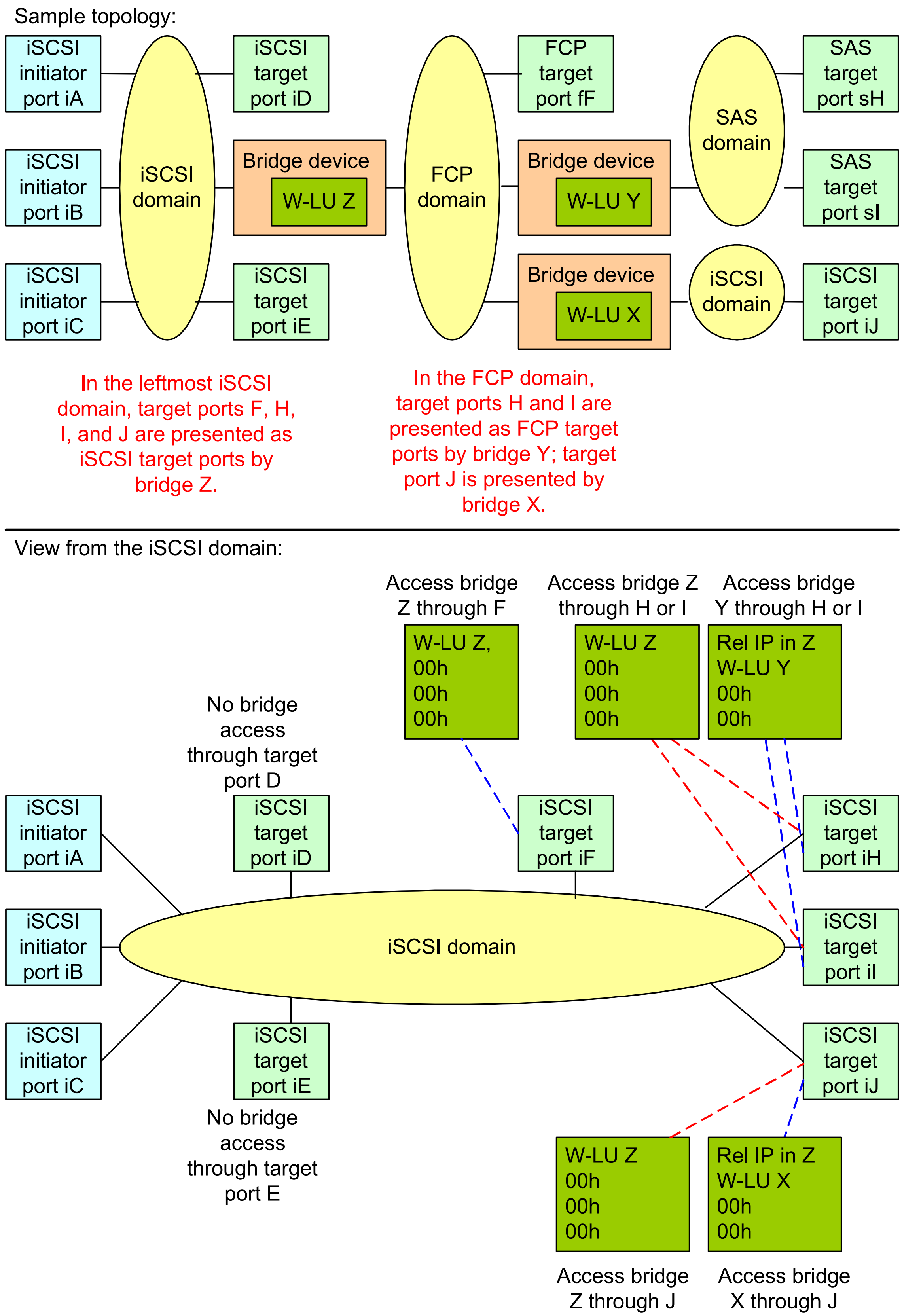


Figure 5 — Nesting and peer bridges

4.3.6 Different initiator port views

Bridges may implement features like zoning, LUN mapping, and LUN masking that present different views of the the domain to different initiator ports. To accommodate this, the REPORT BRIDGE MAPPING command (see 5.2) is a bidirectional command, with an optional parameter list (i.e., write data) specifying which initiator port's view to return.

If the bridge does not employ such features, it returns the same view for every initiator port.

One example is when a backup application needs to obtain the mapping for a copy manager on the far side of a bridge. The application starts by requesting its own initiator port mapping information, learning the true identity of the target ports it is accessing. It then queries the bridges for the mappings for the initiator port(s) used by the copy manager. Combining the two, it is able to determine which target port names/identifiers to use in copy manager target descriptors to describe the correct source and destination.

Figure 6 shows an example of a backup application and a copy manager.

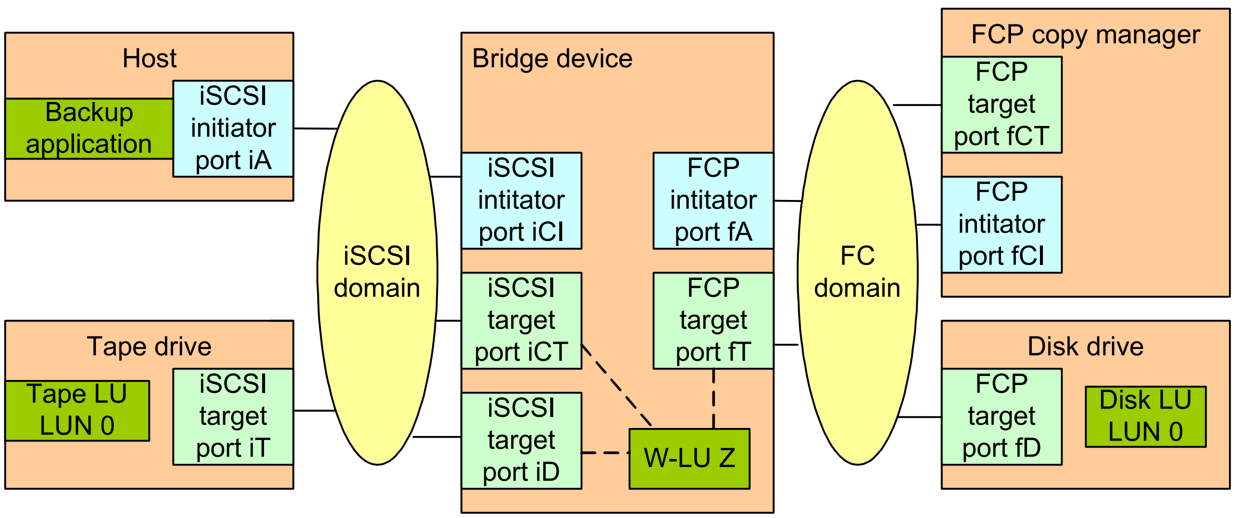


Figure 6 — Copy manager example

Table 9 shows the addresses used in figure 6.

Table 9 — Copy manager addressing

Object	Native address	iSCSI initiator port view	FCP copy manager initiator port view
iSCSI initiator port	(iSCSI name) iA	iA (its own name)	fA (through bridge)
FCP copy manager target port	(FC WWPN) fCT	iCT (through bridge)	fCT
FCP copy manager initiator port	(FC WWPN) fCI	N/A	fCI (its own name)
FC disk drive	(FC WWPN) fD	iD (through bridge)	fD
iSCSI tape drive	(iSCSI name) iT	iT	fT (through bridge)

Table 10 shows the mapping table entries retrieved for iA through iCT or iD.

Table 10 — Mapping table for iA

Object	Near side address	Far side address
FCP copy manager target port	iCT	fCT
FCP disk drive	iD	fD

The backup application discovers that fCT is in a target/initiator device by retrieving the SCSI Ports VPD page (see SPC-3) and noticing its initiator ports. It finds the TransportID for fCI in that VPD page and sends a REPORT BRIDGE MAPPING command specifying fCI as the initiator port whose mapping is requested.

Table 11 shows the mapping table entries retrieved for fCI through iCT or iD.

Table 11 — Mapping table for fCT

Object	Near side address	Far side address
iSCSI tape drive	fT	iT

When it builds a target descriptor for fCI to copy from disk to tape, known to itself as iD and iT, it uses the identifiers that fCI uses for those devices as shown in table 12.

Table 12 — Target descriptors

Object	Value
Source (FCP tape drive) target port identifier	fD
Destination (iSCSI tape drive) target port identifier	fT
LUN	unchanged

4.4 Interceptable commands

4.4.1 Interceptable commands overview

This clause lists commands that present problems for bridges. When the bridge and application client both support this standard, many of these problems are avoided; the bridge just passes through the data to/from the logical unit unchanged and the application knows how to interpret it correctly (for read data) and set it correctly (for write data). To help application clients which do not fully support MSC, particularly the REPORT BRIDGE MAPPING command, the bridge may intercept select commands and make them behave properly.

The REPORT BRIDGE MAPPING command (see 5.2) returns information about which of these commands are intercepted.

4.4.2 INQUIRY command

Device Identification VPD page (i.e., page 83h) identification descriptors (see SPC-3) with an ASSOCIATION field set to 1h (i.e., target port) are based on the far side target port. A transparent bridge has to replace these descriptors with ones representing the near side target port.

Identification descriptors with an IDENTIFIER TYPE field set to 4h (relative target port) are based on the far side target port. A transparent bridge preserves this mapping. A LUN mapping bridge changes this mapping; new relative target port numbers may need to be assigned. This affects commands that use the relative target port identifier (see REPORT/SET TARGET PORT GROUPS and PERSISTENT RESERVE IN/OUT).

Identification descriptors with an ASSOCIATION field set to 2h (i.e., target device) are based on the far side target device. A transparent bridge has to replace these descriptors with ones representing the bridge device itself.

4.4.3 Alias lists

Alias entry designators (see SPC-3) used by the CHANGE ALIASES command and the REPORT ALIASES command are all protocol specific and shall be translated by a transparent bridge.

4.4.4 Extended copy

Most target descriptors (see SPC-3) used by the EXTENDED COPY command and the RECEIVE COPY RESULTS command are protocol specific and shall be translated by a transparent bridge.

4.4.5 Persistent reservations

TransportIDs (see SPC-3) used by the PERSISTENT RESERVE OUT command Specify Initiator Ports feature are protocol specific and shall be translated by a transparent bridge.

Relative target port identifiers returned by the PERSISTENT RESERVE IN command READ FULL STATUS service action (see SPC-3) shall be translated by a transparent bridge.

4.4.6 Access controls commands

TransportIDs (see SPC-3) used by the ACCESS CONTROL IN command and the ACCESS CONTROL OUT command are protocol specific and shall be translated by a transparent bridge.

AccessIDs, on the other hand, should flow through without problems.

4.4.7 Target port group access commands

Relative target port identifiers (see SPC-3) used by the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS command shall be translated by a transparent bridge.

4.4.8 Log pages

The Protocol-Specific log page (see SPC-3) accessed by the LOG SELECT commands and LOG SENSE commands shall be handled by the transparent bridge since the page includes relative target port identifiers.

Since the near and far side meanings can be completely different, these pages should be blocked.

4.4.9 Mode pages

The Protocol-Specific Port mode page (see SPC-3) and Protocol-Specific Logical Unit mode page (see SPC-3) accessed by the MODE SELECT commands and MODE SENSE commands shall be handled by a transparent bridge.

Since the near and far side meanings can be completely different, these pages should be blocked.

4.5 Mapping changes

Mechanisms to change bridge mappings are outside the scope of this standard. The REPORT BRIDGE MAPPING command is only capable of reporting mappings, not changing them.

Changes to the mappings can still occur, however. Application clients may send a WAIT FOR BRIDGE MAPPING CHANGE command (see 5.3) to receive notification via command completion when a mapping change occurs.

4.6 Reservations

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. See SPC-3 for a description of reservations. The details of commands that are allowed under what types of reservations are described in table 13.

Commands from I_T nexuses holding a reservation should complete normally. The behavior of commands from registered I_T nexuses when a registrants only or all registrants type persistent reservation is present is specified in table 13.

For each command, this standard or SPC-3 defines the conditions that result in RESERVATION CONFLICT.

Table 13 — SBC-2 commands that are allowed in the presence of various reservations

Command	Addressed LU has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From I_T nexus not registered	
	Write Exclusive	Exclusive Access		Write Exclusive - RR	Exclusive Access - RR
REPORT BRIDGE MAPPING	Allowed	Allowed	Allowed	Allowed	Allowed
WAIT FOR BRIDGE MAPPING CHANGE	Allowed	Allowed	Allowed	Allowed	Allowed

Key: LU = Logical Unit, RR = Registrants Only or All Registrants
Allowed: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present should complete normally.
Conflict: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present shall not be performed and the device server shall terminate the command with RESERVATION CONFLICT status.

5 Commands for bridge controller devices

5.1 Commands for bridge controller devices overview

The commands for bridge controller devices are listed in table 14.

Table 14 — Commands for bridge controller devices (part 1 of 2)

Command name	Operation code [a]	Type [b]	Reference
ACCESS CONTROL IN	86h	O	SPC-3
ACCESS CONTROL OUT	87h	O	SPC-3
CHANGE ALIASES	A4h/0Bh	O	SPC-3
EXTENDED COPY	83h	O	SPC-3
INQUIRY	12h	M	SPC-3
LOG SELECT	4Ch	O	SPC-3
LOG SENSE	4Dh	O	SPC-3
MODE SELECT (6)	15h	O	SPC-3
MODE SELECT (10)	55h	O	SPC-3
MODE SENSE (6)	1Ah	O	SPC-3
MODE SENSE (10)	5Ah	O	SPC-3
PERSISTENT RESERVE IN	5Eh	O	SPC-3
PERSISTENT RESERVE OUT	5Fh	O	SPC-3
PREVENT ALLOW MEDIUM REMOVAL	1Eh	O	SPC-3
READ ATTRIBUTE	8Ch	O	SPC-3
READ BUFFER	3Ch	O	SPC-3
RECEIVE COPY RESULTS	84h	O	SPC-3
RECEIVE DIAGNOSTIC RESULTS	1Ch	O/M [c]	SPC-3
REPORT ALIASES	A3h/0Bh	O	SPC-3
REPORT BRIDGE MAPPING	A4h/nnh	M	5.2
REPORT DEVICE IDENTIFIER	A3h/05h	O	SPC-3
REPORT LUNS	A0h	M	SPC-3
REPORT PRIORITY	A3h/0Eh	O	SPC-3
REPORT SUPPORTED OPERATION CODES	A3h/0Ch	O	SPC-3
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	A3h/0Dh	O	SPC-3
REPORT TARGET PORT GROUPS	A3h/0Ah	O	SPC-3
REQUEST SENSE	03h	M	SPC-3

Table 14 — Commands for bridge controller devices (part 2 of 2)

Command name	Operation code [a]	Type [b]	Reference
SEND DIAGNOSTIC	1Dh	M	SPC-3
SET DEVICE IDENTIFIER	A4h/06h	O	SPC-3
SET PRIORITY	A4h/0Eh	O	SPC-3
SET TARGET PORT GROUPS	A4h/0Ah	O	SPC-3
TEST UNIT READY	00h	M	SPC-3
WAIT FOR BRIDGE MAPPING CHANGE	A3h/nnh	M	5.3
The following operation codes are vendor-specific: C0h through FFh. All operation codes for bridge controller devices not specified in this table are reserved for future standardization.			
[a] Some commands are defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash. [b] M = command implementation is mandatory. O = command implementation is optional. X = Command implementation requirements detailed in the reference. [c] This command shall be supported if the ENCSERV bit is set to one in the standard INQUIRY data (see SPC-3) and may be supported otherwise.			

Editor's Note 3: SPC-4 must assign service action codes under MAINTENANCE IN for the two commands.

5.2 REPORT BRIDGE MAPPING command

5.2.1 REPORT BRIDGE MAPPING command overview

The REPORT BRIDGE MAPPING command (see table 15) requests information on task management functions (see SAM-3) the addressed logical unit supports. The REPORT BRIDGE MAPPING command is optionally bidirectional:

a) the optional write parameter list lets the application client specify the combination of initiator port and bridge target port for which the bridge shall return mapping table information; and
b) the mandatory read parameter data contains the requested mapping table information.

The REPORT BRIDGE MAPPING command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this standard. The MAINTENANCE

IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the sccs bit equal to one in their standard INQUIRY data (see SPC-3).

Table 15 — REPORT BRIDGE MAPPING command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	Reserved			SERVICE ACTION (nnh)				
2	(MSB)	PARAMETER LIST LENGTH						
5								(LSB)
6	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

The OPERATION CODE field is set to A3h.

The SERVICE ACTION field is set to nnh.

Editor's Note 4: SPC-4 must assign service action codes under MAINTENANCE OUT for the command. Since it is bidirectional, it could also be assigned one under MAINTENANCE IN (A3h).

The PARAMETER LIST LENGTH field specifies the number of bytes of the write parameter list available in the data-out buffer, if any. If no parameter data is provided, the bridge shall return mapping information for the initiator port running the REPORT BRIDGE MAPPING command.

The ALLOCATION LENGTH field specifies the number of bytes that have been allocated for the read parameter data in the data-in buffer. The allocation length shall be at least four. If the allocation length is less than for the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The CONTROL field is defined in SAM-3.

5.2.2 REPORT BRIDGE MAPPING parameter list

The format of the parameter list optionally provided in the data-out buffer is shown in table 16.

Table 16 — REPORT BRIDGE MAPPING parameter list (data-out)

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	RELATIVE TARGET PORT						
1								(LSB)
2	(MSB)	INITIATOR PORT TRANSPORTID LENGTH (n - 3)						
3								(LSB)
4	INITIATOR PORT TRANSPORTID							
n								

The RELATIVE TARGET PORT field specifies the target port through which the specified initiator port applies. A RELATIVE TARGET PORT field set to zero specifes that the target port through which the REPORT BRIDGE MAPPING command was received shall be used.

The INITIATOR PORT TRANSPORTID LENGTH field specifies the length of the INITIATOR PORT TRANSPORTID LENGTH field.

The INITIATOR PORT TRANSPORTID field specifies the TransportID of the initiator port for which the bridge shall return mapping information.

5.2.3 REPORT BRIDGE MAPPING parameter data

The format of the parameter data returned in the data-in buffer is shown in table 17.

Table 17 — REPORT BRIDGE MAPPING parameter data (data-in)

Byte\Bit	7	6	5	4	3	2	1	0
0	EXTENDED COPY	ACCESS CONTROL	PERSISTENT RESERVE	TARGET PORT GROUPS	ALIAS	MODE	LOG	INQUIRY
1	Reserved							
3								
4	(MSB)	MAPPING TABLES LENGTH (m - 7)						
7								(LSB)
	Mapping table entries							
8	Mapping table entry (first)							
	...							
	Mapping table entry (first)							
m								

An EXTENDED COPY bit set to one means the bridge intercepts the EXTENDED COPY command (see 4.4.4). An EXTENDED COPY bit set to one means the bridge does not intercept the EXTENDED COPY command.

An ACCESS CONTROL bit set to one means the bridge intercepts the ACCESS CONTROL IN command and the ACCESS CONTROL OUT command (see 4.4.6). An ACCESS CONTROL bit set to one means the bridge does not intercept the ACCESS CONTROL IN command and the ACCESS CONTROL OUT command.

A PERSISTENT RESERVE bit set to one means the bridge intercepts the PERSISTENT RESERVE IN command and the PERSISTENT RESERVE OUT command (see 4.4.5). A PERSISTENT RESERVE bit set to one means the bridge does not intercept the PERSISTENT RESERVE IN command and the PERSISTENT RESERVE OUT command.

A TARGET PORT GROUPS bit set to one means the bridge intercepts the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS command (see 4.4.7). A TARGET PORT GROUPS bit set to one means the bridge does not intercept the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS command.

An ALIAS bit set to one means the bridge intercepts the CHANGE ALIASES command and the REPORT ALIASES command (see 4.4.3). An ALIAS bit set to one means the bridge does not intercept the CHANGE ALIASES command and the REPORT ALIASES command.

A MODE bit set to one means the bridge intercepts the MODE SENSE commands and the MODE SELECT commands (see 4.4.9). A MODE bit set to one means the bridge does not intercept the MODE SENSE commands and the MODE SELECT commands.

A LOG bit set to one means the bridge intercepts the LOG SENSE commands and the LOG SELECT commands (see 4.4.8). A LOG bit set to one means the bridge does not intercept the LOG SENSE commands and the LOG SELECT commands.

An INQUIRY bit set to one means the bridge intercepts the INQUIRY command (see 4.4.2). An INQUIRY bit set to one means the bridge does not intercept the INQUIRY command.

The mapping table indicates:

a) how the bridge is mapping far side target ports and logical units into near side target ports and logical units as seen by the initiator port specified in the write parameter list; and
b) how the bridge is mapping the specified initiator port into a far side initiator port as seen by each of those far side target ports and logical units.

The MAPPING TABLES LENGTH field indicates the length of all the mapping table entries that follow. The bridge shall return all mapping table entries applicable to the specified initiator port. If the allocation length is not large enough to return all the mapping table entries, this field shall not be changed.

Table 18 defines the mapping table entry format.

Table 18 — Mapping table entry

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	ENTRY LENGTH (n - 1)						
1								(LSB)
2	(MSB)	NEAR SIDE RELATIVE TARGET PORT						
3								(LSB)
4		NEAR SIDE LOGICAL UNIT NUMBER						
11								
12	(MSB)	FAR SIDE RELATIVE INITIATOR PORT						
13								(LSB)
14	(MSB)	FAR SIDE TARGET DESCRIPTOR LENGTH (n - 15)						
15								(LSB)
16		FAR SIDE TARGET DESCRIPTOR						
n								

The ENTRY LENGTH field indicates the number of bytes that that follow in the mapping table entry.

The NEAR SIDE RELATIVE TARGET PORT field contains the relative target port identifier of the bridge device capable of mapping to one or more logical units.

The NEAR SIDE LOGICAL UNIT NUMBER field shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped. Otherwise, it contains the logical unit number accessible through the target port identified by the NEAR SIDE RELATIVE TARGET PORT field/

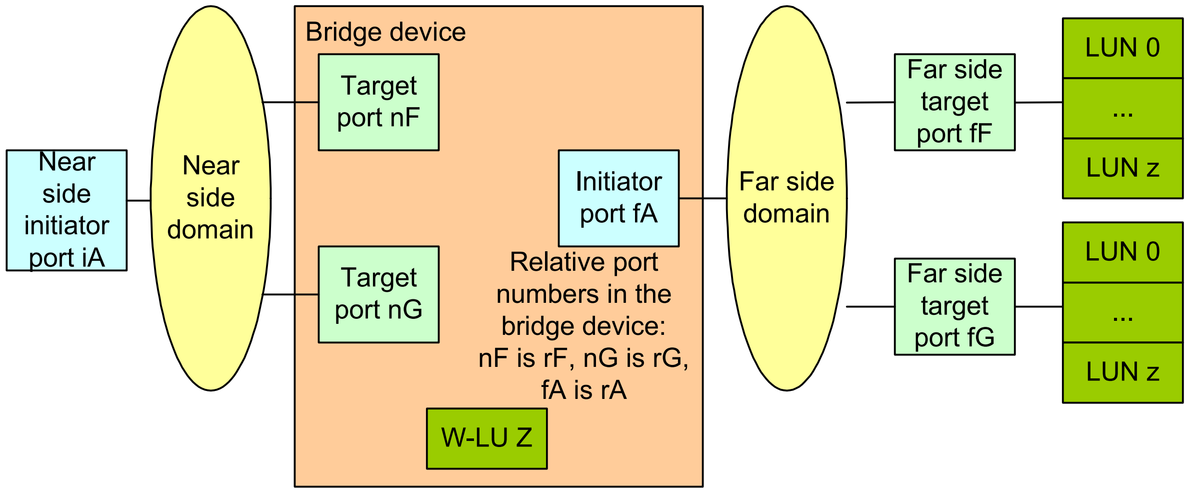
The BACK SIDE INITIATOR TRANSPORTID LENGTH field indicates the number of bytes in the BACK SIDE INITIATOR TRANSPORTID field.

The FAR SIDE RELATIVE INITIATOR PORT field contains the relative port identifier of the bridge device’s initiator port used to access the specified far side target port and logical unit(s).

The FAR SIDE TARGET DESCRIPTOR LENGTH field indicates the number of bytes in the BACK SIDE TARGET DESCRIPTOR field.

The FAR SIDE TARGET DESCRIPTOR field contains an EXTENDED COPY target descriptor (see SPC-3) identifying a target port and one or more logical units that a near side initiator port accesses through the target port indicated by the NEAR SIDE RELATIVE TARGET PORT field. The bridge forwards these accesses with the initiator port indicated by the FAR SIDE RELATIVE INITIATOR PORT field. The LU IDENTIFIER field of the far side target descriptor shall be set to FFFFFFFF FFFFFFFFh if all logical units behind the target port are being mapped.

Figure 7 shows an example of a mapping table.



When iA queries W-LU Z through iF, the mapping table returns:
a) near side relative target port rF;
b) near side LUN=FFFFFFFF FFFFFFFFh;
c) far side relative initiator port rA; and
d) far side target descriptor describing target port fF with LUN=FFFFFFFF FFFFFFFFh

When iA queries W-LU Z through iG, the mapping table returns:
a) near side relative target port rG;
b) near side LUN=FFFFFFFF FFFFFFFFh;
c) far side relative initiator port rA; and
d) far side target descriptor describing target port fF with LUN=FFFFFFFF FFFFFFFFh

Figure 7 — Mapping table example

5.3 WAIT FOR BRIDGE MAPPING CHANGE command

The WAIT FOR BRIDGE MAPPING command (see table 19) waits for a bridge mapping to change before completing. After this command completes, an application client should send a REPORT BRIDGE MAPPING command (see 5.2) to determine what changed.

The WAIT FOR BRIDGE MAPPING CHANGE command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this standard. The MAINTENANCE IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the sccs bit equal to one in their standard INQUIRY data (see SPC-3).

Table 19 — WAIT FOR BRIDGE MAPPING TABLE command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (nnh)				
2	Reserved							
9								
10	Reserved							
11	CONTROL							

The OPERATION CODE field is set to A3h.

The SERVICE ACTION field is set to nnh.

Editor’s Note 5: SPC-4 must assign service action codes under MAINTENANCE IN for the command.

The CONTROL field is defined in SAM-3.

6 Parameters for bridge controller devices

6.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with bridge controller devices. The diagnostic page codes for bridge controller are defined in table 20.

Table 20 — Diagnostic page codes

Diagnostic page code	Description	Reference
00h	Supported diagnostic pages	SPC-3
01h - 2Fh	SCSI enclosure services diagnostic pages	SES-2
30h - 3Fh	Diagnostic pages assigned by SPC-3	SPC-3
40h - 7Fh	Reserved for this standard	
80h - FFh	Vendor-specific diagnostic pages	

6.2 Log parameters

This subclause defines the descriptors and pages for log parameters used with bridge controller devices. See SPC-3 for a detailed description of logging operations. The log page codes for bridge controller devices are defined in table 21.

Table 21 — Log page codes

Log page code	Description	Reference
00h	Supported Log Pages log page	SPC-3
01h	Buffer Over-Run/Under-Run log page	SPC-3
02h	Write Error Counter log page	SPC-3
03h	Read Error Counter log page	SPC-3
04h	Reserved	
05h	Verify Error Counter log page	SPC-3
06h	Non-Medium Error log page	SPC-3
07h	Last n Error Events log page	SPC-3
08h	Reserved	
09h	Restricted (see SPC-3)	
0Ah	Restricted (see SPC-3)	
0Bh	Last n Deferred Errors Or Asynchronous Events log page	SPC-3
0Ch	Reserved	
0Dh	Temperature log page	SPC-3
0Eh	Start-Stop Cycle Counter log page	SPC-3
0Fh	Application Client log page	SPC-3
10h	Self-Test Results log page	SPC-3
11h - 17h	Reserved	
18h	Protocol-Specific Port log page	SPC-3
19h - 2Eh	Reserved	
2Fh	Informational Exceptions log page	SPC-3
30h - 3Eh	Vendor-specific	
3Fh	Reserved	

6.3 Mode parameters

This subclause defines the block descriptors and mode pages used with bridge controller devices.

The mode parameter list, including the mode parameter header, is described in SPC-3.

The MEDIUM TYPE field in the mode parameter header (see SPC-3) is reserved for bridge controller devices.

The DEVICE-SPECIFIC PARAMETER field in the mode parameter header (see SPC-3) is reserved for bridge controller devices.

The BLOCK DESCRIPTOR LENGTH field in the mode parameter header (see SPC-3) shall be set to zero. Bridge controller devices do not support mode parameter block descriptors.

The mode page codes and subpage codes for bridge controller devices are shown in table 22.

Table 22 — Mode page codes for bridge controller devices

Mode page code	Description	Reference
00h	Vendor-specific (does not require page format)	
00h-3Eh/FFh	Return all subpages [a]	SPC-3
01h		
02h	Disconnect-Reconnect mode page	SPC-3
03h - 09h	Reserved	
0Ah/00h	Control mode page	SPC-3
0Ah/01h	Control Extension mode page	SPC-3
0Ah/02h - 3Eh	Reserved	
0Bh - 13h	Reserved	
14h	Enclosure Services Management mode page [b]	SES-2
15h - 17h	Reserved	
18h	Protocol-Specific LUN mode page	SPC-3
19h	Protocol-Specific Port mode page	SPC-3
1Ah	Power Condition mode page	SPC-3
1Bh	Reserved	
1Ch	Informational Exceptions Control mode page	SPC-3
1Dh - 1Fh	Reserved	
20h - 3Eh	Vendor-specific (does not require page format)	
3Fh/00h	Return all mode pages [a]	SPC-3
3Fh/01h - 3Eh	Reserved	
3Fh/FFh	Return all mode pages and subpages [a]	SPC-3

[a] Valid only for the MODE SENSE command
[b] Valid only if the ENCSERV bit is set to one in the standard INQUIRY data (see SPC-3)

Annex A
(informative)

Numeric order codes

A.1 Service action CDBs

Some commands in table 14 (see 5.1) are implemented as device-type specific service actions for the MAINTENANCE IN operation code A3h (see SPC-3). These commands are differentiated by service action codes as described in table A.1.

Table A.1 — MAINTENANCE IN service actions

Operation code/service action code	Description
A3h/nnh	WAIT FOR BRIDGE MAPPING CHANGE

Some commands in table 14 (see 5.1) are implemented as device-type specific service actions for the MAINTENANCE OUT operation code A4h (see SPC-3). These commands are differentiated by service action codes as described in table A.2.

Table A.2 — MAINTENANCE OUT service actions

Operation code/service action code	Description
A4h/nnh	REPORT BRIDGE MAPPING

Editor’s Note 6: SPC-4 must assign the opcodes. REPORT BRIDGE MAPPING could use A3h instead (it is bidirectional).